

**Программно-аппаратная платформа для контроля мест накопления отходов  
и управления логистикой отходов, в том числе твердых коммунальных  
отходов (ТКО)**

**Общее описание системы**

**Лист утверждения**

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата
	 29.05.2020			

Москва 2020

## СОДЕРЖАНИЕ

<b>ПЕРЕЧЕНЬ ОПРЕДЕЛЕНИЙ И СОКРАЩЕНИЙ.....</b>	<b>4</b>
<b>1 НАЗНАЧЕНИЕ СИСТЕМЫ.....</b>	<b>5</b>
1.1. Вид деятельности, для автоматизации которой предназначена система.....	5
1.2. Объекты автоматизации .....	5
1.3. Перечень функций, выполняемых системой .....	6
<b>2. ОПИСАНИЕ СИСТЕМЫ.....</b>	<b>7</b>
2.1. Функциональная структура системы.....	7
2.2. Сведения о системе, необходимые для эксплуатации .....	10
2.3. Описание функционирования системы и ее частей .....	10
2.3.1. Описание функций модуля ведения нормативно-справочной информации .....	10
2.3.2. Функциональный блок управления местами сбора и накопления отходов и образователями отходов.....	11
2.3.3. Функциональный блок ведения справочников и классификаторов .....	11
2.3.4. Функциональный блок ведения сотрудников .....	11
2.3.5. ФБ ведения справочных данных об АТО .....	11
2.3.6. ФБ ведения справочных данных о местах сбора отходов .....	11
2.3.7. ФБ ведения справочных данных о местах разгрузки .....	12
2.3.8. Описание функций модуля ведения договоров .....	12
2.3.10. Описание функций модуля управление заказом (сбор заявок).....	13
2.3.11. Функциональный блок ведения договоров вывоза .....	13
2.3.12. Функциональный блок ведения заявок.....	14
2.3.13. Функциональный блок работы с картой площадок.....	14
2.3.14. Функциональный блок работы с графиками вывоза .....	14
2.3.15. Функциональный блок «Личный кабинет Заказчика вывоза отходов» .....	15
2.3.16. Описание функций модуля контроля и учета вывоза .....	15
2.3.17. ФБ Взаимодействия с мусоровывозящими организациями .....	15
2.3.18. Функции ведения договоров управления вывозом.....	16
2.3.19. Функция работы с механизмами распределения зон деления возчиков .....	16
2.3.20. Функция передачи площадок и заявок возчикам, согласно зонам ответственности	17
2.3.21. Функция сбора и визуализации информации по фактам вывоза возчиками .....	17
2.3.22. Функция получения информации о ТС возчика .....	17
2.3.23. Функция тестирования навесного оборудования ТС возчика.....	17
2.3.24. Функция согласования ТС возчика для последующего использования в вывозе заданий РО.....	18
2.3.25. Функция контроля выполнения заданий возчиком .....	18
2.3.26. Функция дополнительных режимов контроля.....	18
2.3.27. Описание функций модуля аналитики и отчетности .....	18
2.3.28. Функциональный блок формирования отчетов .....	18
2.3.29. Функциональный блок аналитической отчетности.....	19
2.3.30. Описание функций модуля пользовательского интерфейса .....	19
2.3.31. Описание функций модуля администрирования .....	20
2.3.32. Описание функций модуля аудита.....	20

<b>3. ВЗАИМОСВЯЗЬ СО СМЕЖНЫМИ СИСТЕМАМИ.....</b>	<b>21</b>
<b>3.1. Общие сведения.....</b>	<b>21</b>
<b>3.2. Функция телеметрии.....</b>	<b>21</b>
<b>3.3. Функция взаимодействия с внешними информационными системами ведения контрагентов, договоров и расчета заработной платы .....</b>	<b>21</b>
<b>3.4. Функция взаимодействия с внешними информационными системами оператора датчиков .....</b>	<b>22</b>
<b>Приложение №1 Описание информационного взаимодействия с внешней системой в части передачи информации по договорам и площадкам.....</b>	<b>23</b>
<b>Приложение №2. Описание Public API системы контроля датчиков .....</b>	<b>26</b>

## ПЕРЕЧЕНЬ ОПРЕДЕЛЕНИЙ И СОКРАЩЕНИЙ

Сокращение	Определение
АИС	Автоматизированная информационная система
АРМ	Автоматизированное рабочее место
БНСО	Бортовое навигационно-связное оборудование
ЖКХ	Жилищно-коммунальное хозяйство
РФ	Российская Федерация
СУБД	Система управления базами данных
ТКО	Твердые коммунальные отходы
ТТН	Товарно-транспортная накладная
ТС	Транспортное средство
ФБ	Функциональный блок
ФЗ	Федеральный закон
Датчик	Аппаратный компонент платформы АИС «Отходы» производства ООО «Эттон-Центр» для измерения уровня заполненности емкостей, контролю температуры и опрокидывания контейнера и передачи информации на сервер по беспроводному GSM-каналу
ПАП, Система	Программно-аппаратная платформа, включающий систему АИС «Отходы», датчики, систему интеграции датчиков и АИС «Отходы»
1С ООО «УК ПЖКХ»	Учетная система ООО «УК ПЖКХ» на платформе «1С:Предприятие 8»
1С ООО «УК ПЖКХ» (ФОТ)	Учетная система для расчета заработной платы ООО «УК ПЖКХ» на платформе «1С:Предприятие 8»

# **1 НАЗНАЧЕНИЕ СИСТЕМЫ**

## **1.1. Вид деятельности, для автоматизации которой предназначена система**

Программно-аппаратная платформа (ПАП), включающей программный компонент АИС «Отходы», разработанный ООО «ТКО-Информ» и аппаратный компонент - ультразвуковые датчики производства ООО «Эттон-Центр».

АИС «Отходы» представляет собой комплексное решение, предназначенное для обеспечения сквозной автоматизации деятельности в сфере обращения с отходами. Областями применения АИС «Отходы» является комплекс задач автоматизации деятельности в сфере обращения с отходами, включая:

- формирование единого информационного и технологического пространства в сфере обращения с отходами;
- автоматизация функций государственного контроля и надзора в интересах органов государственной власти субъекта РФ;
- автоматизация деятельности регионального оператора по обращению с отходами в соответствии с требованиями Федерального закона от 24.06.1998 N 89-ФЗ «Об отходах производства и потребления»;
- информационно-аналитическая поддержка органов местного самоуправления в части реализации контроля процессов обращения с отходами;
- автоматизация деятельности организаций, задействованных в сфере обращения с отходами.

Использование УЗ датчиков для размещения на контейнерах для сбора ТКО и транспортных средствах позволяет осуществлять эффективный и непрерывный контроль процессов накопления и транспортирования ТКО на всех этапах логистической цепочки. Конкурентные решения, представленные на рынке, по причинам технологического характера не обеспечивают возможности сбора первичной информации со сравнимым уровнем полноты и детализации.

Интеграция программных и аппаратных средств, входящих в состав ПАП, позволяет обеспечивать повышение эффективности логистики ТКО за счет, исключения возможности намеренных нарушений в процессе обращения с ТКО, оптимизации логистических схем транспортировки и переработки ТКО.

## **1.2. Объекты автоматизации**

Объектами автоматизации АИС «Отходы» являются:

- региональные операторы по обращению с отходами;
- заказчики услуг сбора, транспортирования, утилизации и размещения отходов;

- организации, осуществляющие деятельность по сбору, накоплению, транспортированию, утилизации и размещению отходов.

### **1.3. Перечень функций, выполняемых системой**

Для реализации поставленных целей система в процессе своего функционирования должна решать следующие задачи:

- организация и контроль производственной деятельности регионального оператора;
- управление заказами на предоставление услуг сбора, транспортирования, утилизации и размещения отходов;
- планирование работ по транспортированию, утилизации и размещению отходов;
- оперативное управление процессами обращения с отходами;
- контроль накопления, сбора, транспортирования, утилизации и размещения отходов, включая контроль деятельности регионального оператора по обращению с отходами;
- мониторинг наполняемости контейнеров на площадках накопления отходов (контейнерные площадки);
- реализация типовых интерфейсов для обеспечения возможности информационного взаимодействия с иными информационными системами в рамках предоставления услуг по транспортированию, переработке и размещению отходов, включая поддержку открытого API для расширения сферы функционального применения АИС «Отходы»;
- информационное взаимодействие пользователей в рамках предоставления услуг по сбору, транспортированию, переработке и размещению отходов;
- информационная поддержка управления функционированием отрасли обращения с отходами;
- формирование и ведение базы нормативно-справочной информации, необходимой для автоматизации процессов отрасли обращения с отходами;
- формирование объективной аналитической отчетности по результатам контроля процессов функционирования отрасли обращения с отходами;
- измерение объема ТКО в контейнере и грузовой ёмкости транспортного средства;
- фиксация данных по посещению площадки и сбору ТКО на площадке;
- отправка собранных данных на сервер;
- передача геолокационных данных;
- сбор данных по состоянию контейнерной площадки (предупреждение об аварийных ситуациях на площадке – мониторинг возгораний в контейнерах; мониторинг опрокидывания контейнеров).

## 2. ОПИСАНИЕ СИСТЕМЫ

### 2.1. Функциональная структура системы

Система АИС «Отходы» представляет собой единое комплексное полнофункциональное решение, обеспечивающее информационно-технологическую поддержку и контроль функционирования и полносвязного взаимодействия участников отрасли обращения с отходами. Система обеспечивает автоматизацию деятельности регионального оператора ТКО по организации обращения с отходами, предоставление информационных сервисов для всех участников отрасли обращения с отходами, а также информационное и функциональное взаимодействие участников отрасли между собой в рамках предоставления услуг в сфере обращения с отходами. Для этого система строится на основе централизованной модели информационного взаимодействия участников (рисунок 1).

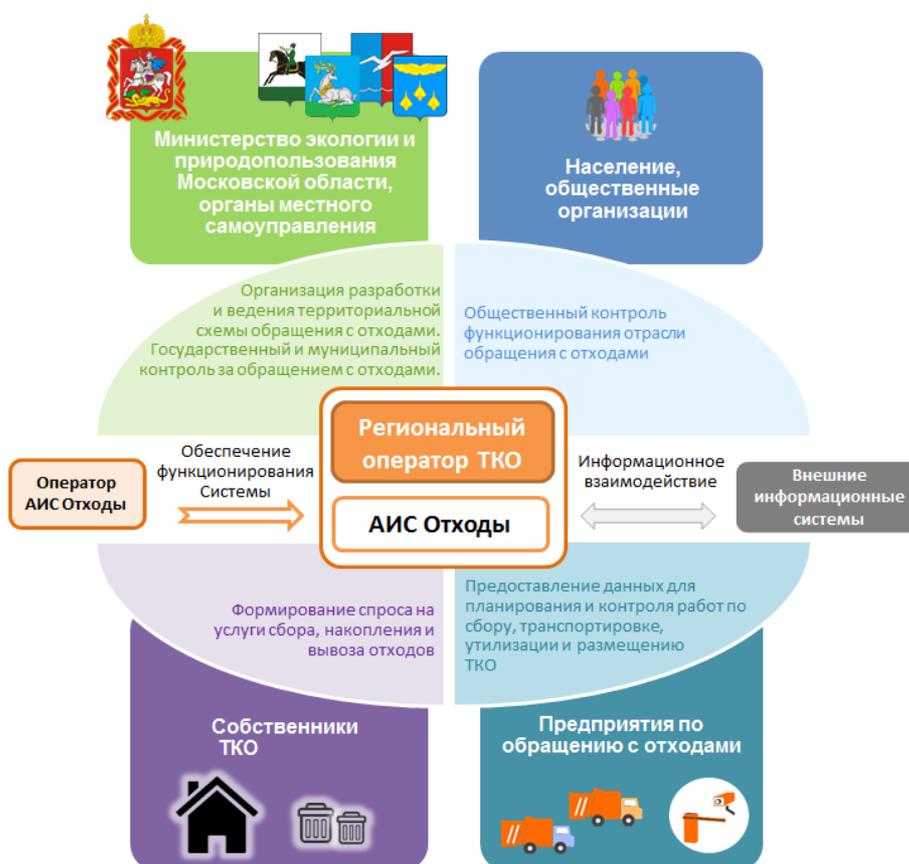


Рисунок 1. Модель информационного взаимодействия участников отрасли обращения с отходами с использованием АИС «Отходы»

Функционирование системы основано на комплексном использовании современных средств информатики, вычислительной техники, телекоммуникационных и навигационных технологий, обеспечивающих учет объектов образования, сбора, накопления, транспортировки, утилизации и размещения отходов, мониторинг и передачу данных от транспортных средств,

осуществляющих перевозку отходов, обмен оперативными данными между элементами и структурными компонентами системы.

Архитектура системы приведена на следующем рисунке:

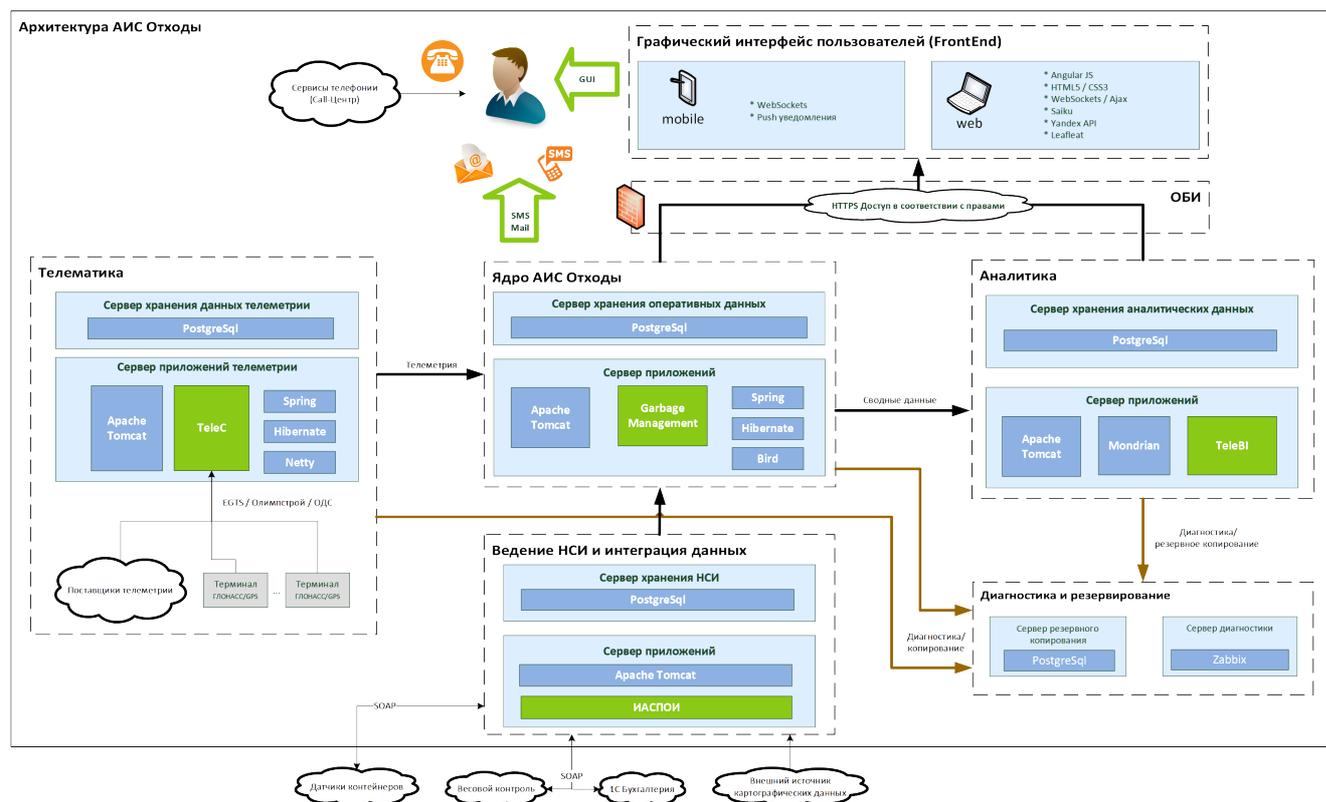


Рисунок 2. Архитектура системы

В составе АИС «Отходы» реализованы следующие функциональные модули:

- **Модуль ведения нормативно-справочной информации**, предназначенный для формирования и поддержания в актуальном состоянии единой базы нормативно-справочной, конфигурационной и архивной информации, необходимых для автоматизации процессов планирования и контроля в сфере обращения с отходами.
- **Модуль ведения договоров**, предназначен для ведения реестра контрагентов, реестра договоров с контрагентами.
- **Модуль управление заказом (сбор заявок)**, предназначен формирования перечня заявок от заказчика вывоза ТКО на оказание услуг сбора и транспортировки ТКО, выставление счетов по вывозу.
- **Модуль контроля и учета вывоза в составе функциональных блоков**:
  - o **Функциональный блок взаимодействия с мусоровывозящими организациями**, предназначенная для передачи заявок, площадок, на сбор и транспортировку ТКО. Контроль исполнения услуг по вывозу и размещению отходов. Выполнение взаиморасчетов с возчиком.

- **Функциональный блок интеграции с внешними системами**, предназначенная для обеспечения информационного взаимодействия АИС «Отходы» с внешними информационными системами, а также окончательным оборудованием мониторинга процессов обращения с отходами.
- **Функциональный блок «Телематическая платформа»**, предназначенная для приема навигационной мониторинговой информации от бортового навигационно-связного оборудования (БНСО) и датчиков ТС организаций, осуществляющих сбор и транспортирование отходов.
- **Модуль аналитики и отчетности**, предназначенная для аналитической и статистической обработки результатов контроля обращения с отходами, расчета макропоказателей, анализа трендов, формирования прогнозов, регламентированных и нерегламентированных отчетных форм по функционированию отрасли обращения с отходами.
- **Модуль пользовательского интерфейса**, предназначенная для обеспечения непосредственного доступа пользователей к Системе в соответствии с правами доступа, определяемыми их должностными обязанностями и задачами.
- **Модуль администрирования**, предназначенная для настройки параметров функционирования АИС «Отходы», параметров доступа к системе, включая информационное взаимодействие с внешними системами, протоколирования действий пользователей.
- **Модуль аудита**, предназначенная для фиксации основных событий в системе для дальнейшего использования в разборе возможных инцидентов.
- **Модуль контроля и управления доступом** транспортных средств по перевозке отходов на объекты обработки и размещения отходов;
- **Модуль автоматизированного согласования объемов отходов** (с использованием электронного документооборота);
- **Модуль сводного контроля состояния** и прогнозирования заполнения объекта утилизации отходов;
- **Модуль автоматизации планирования вывоза** отходов (создания маршрутных и путевых листов);
- **Модуль информационной поддержки экипажей**, осуществляющих вывоз отходов (электронное маршрутное задание, маршрутная навигация);
- Мобильное приложение для водителей.

## **2.2. Сведения о системе, необходимые для эксплуатации**

Работа по управлению ТБО ведется в системе АИС «Отходы», представленной сервисом со стороны ООО «ТКО-Информ». Хостинг для системы, разворачивание и администрирование системы обеспечивает ООО «ТКО-Информ».

Входящие в рабоче-конструкторскую документацию к системе руководства пользователя позволяют получить достаточные для работы в системе знания.

АИС «Отходы» позволяет обеспечить непрерывный сбор и автоматизированную обработку данных об объеме, статусе и местоположении ТКО на всех этапах логистической цепочки.

Структурно платформа включает в себя программные компоненты в составе:

- Автоматизированная информационная система по учету и контролю размещения отходов на инфраструктурных объектах (АИС «Отходы-Инфраструктура»);
- Автоматизированная информационная система по управлению производственной деятельностью предприятий-операторов в сфере обращения с отходами (АИС «Отходы-Региональный оператор»);
- Автоматизированная информационная система по управлению предприятиями жилищно-коммунального хозяйства по вывозу и утилизации отходов (АИС «Отходы-Транспортирование»)

Размещение многоканальных УЗ-датчиков производства компании «Эттон-Центр» на контейнерах сбора ТКО, передача и обработка данных с датчиков в системе АИС Отходы позволяет обеспечить непрерывный контроль заполнения емкостей и корректировать, в зависимости от заполненности контейнеров на площадке, процесс сбора и транспортировки ТКО.

Встроенный датчик возгорания передает в АИС «Отходы» сигналы о тлении или горении ТКО, что позволяет принять меры по предотвращению утраты контейнера.

## **2.3. Описание функционирования системы и ее частей**

### **2.3.1. Описание функций модуля ведения нормативно-справочной информации**

Модуль ведения нормативно-справочной информации включает в себя следующие функциональные блоки (Далее - ФБ):

- ФБ управления местами сбора и накопления отходов и образователями отходов.
- ФБ ведения справочников и классификаторов.
- ФБ ведения сотрудников;
- ФБ ведения справочных данных об АТО;
- ФБ ведения справочных данных о местах сбора отходов;
- ФБ ведения справочных данных о местах разгрузки.

### **2.3.2. Функциональный блок управления местами сбора и накопления отходов и образозателями отходов**

ФБ управления местами сбора и накопления отходов и образозателями отходов предназначен для учета данных о местах сбора и накопления ТКО и образозателях отходов.

ФБ управления местами сбора и накопления отходов и образозателями отходов обеспечивает выполнение следующих функций:

- Ведение реестра мест сбора и накопления отходов;
- Ведение реестра образозателей отходов.

### **2.3.3. Функциональный блок ведения справочников и классификаторов**

ФБ ведения справочников и классификаторов предназначен для управления наполнением справочников и классификаторов, используемых Системой.

ФБ ведения справочников и классификаторов обеспечивает выполнение следующих функций:

- Добавление новой записи в справочник;
- Редактирование записи справочника;
- Удаление записи справочника.

### **2.3.4. Функциональный блок ведения сотрудников**

ФБ ведения сотрудников предназначен для кадрового учета сотрудников.

ФБ ведения сотрудников должен обеспечивать выполнение следующих функций:

- Ведение данных о сотрудниках;
- Установка видимости районов сотрудниками;
- Назначение роли сотрудника.

### **2.3.5. ФБ ведения справочных данных об АТО**

Работа с ФБ осуществляется через меню «НСИ – Редактор районов» и поддерживает следующие функции:

- Создание справочной записи для отдельно взятого АТО.
- Изменение справочной записи для отдельно взятого АТО.

Вся совокупность записей об АТО имеет древовидную структуру: для каждого АТО задается собственное наименование и АТО, в который включено данное. Верхнеуровневые АТО включены в абстрактное АТО «ВСЕ». В случае необходимости можно изменить как наименование АТО, так и ссылку на АТО родительское по отношению к нему.

### **2.3.6. ФБ ведения справочных данных о местах сбора отходов**

Работа с ФБ осуществляется через меню «Реестры – Реестры площадок» и поддерживает следующие функции:

- Создание отдельно взятого места сбора отходов (далее – Площадка).

- Изменение Площадки.
- Удаление Площадки.

Для каждой Площадки требуется указание следующих параметров:

- Место расположения (географические координаты).
- Адрес.
- Административно-территориальное образование, в котором расположена Площадка.

Дополнительно могут быть указаны состав и типы емкостей для сбора отходов, установленных на Площадке и вид отходов.

### **2.3.7. ФБ ведения справочных данных о местах разгрузки**

Работа с ФБ осуществляется через меню «НСИ – Редактор мест разгрузки» и поддерживает следующие функции:

- Создание отдельно взятого места разгрузки.
- Изменение места разгрузки.
- Удаление места разгрузки.

Для каждого места разгрузки требуется указание следующих параметров:

- Тип (МСС, МПС, МПП, полигон).
- Наименование.
- Адрес.
- Место расположения (географические координаты).

Могут быть указаны дополнительные плановые показатели места разгрузки:

- Дата ввода в эксплуатацию.
- Дата вывода из эксплуатации.
- Предельная масса (объем) отходов (для полигонов).
- Предельный суточный объем (для перерабатывающих объектов).

Также может быть указана метка (тэг) для включения места разгрузки в условную группу. Используется при ограничении вывода мест разгрузки на форме визуализации данных.

### **2.3.8. Описание функций модуля ведения договоров**

Модуль ведения договоров включает в себя следующие функциональные блоки (Далее - ФБ):

- ФБ управления заказчиками вывоза ТКО и договорами.

### **2.3.9. Функциональный блок управления заказчиками вывоза ТКО**

ФБ управления заказчиками вывоза ТКО предназначен для ведения информации по заказчикам вывоза отходов, осуществляющих подачу заявок на вывоз ТКО.

ФБ управления заказчиками вывоза ТКО обеспечивает выполнение следующих функций:

- Добавление информации о заказчике;
- Редактирование информации о заказчике;
- Смена статуса обслуживания заказчика;
- Настройка политики заказов;
- Настройка особых отметок заказчика.

### **2.3.10. Описание функций модуля управление заказом (сбор заявок)**

Модуль управление заказом (сбор заявок) включает в себя следующие функциональные блоки:

- ФБ ведения договоров вывоза.
- ФБ ведения заявок.
- ФБ работы с картой площадок.
- ФБ работы с графиками вывоза.
- ФБ «Личный кабинет заказчика вывоза отходов»

### **2.3.11. Функциональный блок ведения договоров вывоза**

ФБ предназначен для ведения договорных отношений между заказчиком вывоза и региональным оператором.

ФБ ведения договоров вывоза обеспечивает выполнение следующих функций:

- Создание договора вывоза с формированием набора атрибутов договора, таких как:
  - идентификационные данные договора:
    - номер;
    - дата подписания;
    - срок действия.
  - данные о контрагентах договора:
    - наименование;
    - вид услуг/работ;
    - тип отходов;
    - вид тарифа;
    - лимиты;
    - вид договора;

- прототипы;
- объем вывоза;
- перечень площадок.
- сопутствующие документы (сканы оригиналов сопутствующих документов);
- дополнительные соглашения.
- Редактирование договора вывоза;
- Удаление договора вывоза.

### **2.3.12. Функциональный блок ведения заявок**

ФБ ведения заявок предназначен для формирования перечня заявок, полученных от заказчика вывоза ТКО на оказание услуг сбора и транспортировки ТКО.

ФБ ведения заявок должен обеспечивать выполнение следующих функций:

- Отображение перечня заявок по каждому заказчику вывоза ТКО;
- Фильтрация заявок заказчика вывоза ТКО по атрибутам;
- Создание заявки на вывоз ТКО;
- Создание групповой заявки на вывоз ТКО;
- Создание заявки на подбор ТКО;
- Копирование заявки;
- Редактирование заявки;
- Удаление заявки;
- Печать заявок на вывоз ТКО;
- Печать истории заявок на вывоз ТКО;
- Формирование заявок в соответствии с графиком вывоза.

### **2.3.13. Функциональный блок работы с картой площадок**

ФБ предназначен для отображения информации по площадкам на карте.

ФБ работа с картой площадок обеспечивает выполнение следующих функций:

- Отображение площадок на карте;
- Отображение точек разгрузки на карте;
- Отображение стоянок транспортных средств на карте;
- Отображение сводной информации по установленным на площадках ёмкостях и результатам инвентаризации площадок.

### **2.3.14. Функциональный блок работы с графиками вывоза**

ФБ предназначен для ведения реестра графиков вывоза ТКО с мест сбора и накопления отходов.

ФБ работы с графиками вывоза обеспечивает выполнение следующих функций:

- создание графика вывоза с места сбора и накопления отходов;
- редактирование графика вывоза с места сбора и накопления отходов;
- удаление графика вывоза с места сбора и накопления отходов.

При создании и редактировании графика вывоза с места сбора и накопления отходов пользователь имеет возможность указать следующие данные:

- тип графика вывоза (произвольный, недельный);
- наименование графика вывоза;
- срок действия графика вывоза;
- действие с емкостью;
- период вывоза каждой емкости;
- количество емкостей для вывоза.

### **2.3.15. Функциональный блок «Личный кабинет Заказчика вывоза отходов»**

ФБ предназначен подачи заявок заказчиком вывоза ТКО, корректировки по заявкам, подачи жалоб на оказание услуг.

ФБ «Личный кабинет Заказчика вывоза отходов» обеспечивает выполнение следующих функций:

- Отображение перечня заявок;
- Создание заявки на вывоз ТКО;
- Редактирование заявки на вывоз ТКО;
- Удаление заявки на вывоз ТКО;
- Регистрация жалобы на ненадлежащее качество оказания услуги;
- Ввод дополнительных данных для подтверждения обоснованности жалобы;
- Подача претензии по вывозу;
- Согласование вывоза;
- Изменение поданной претензии;
- Отзыв поданной претензии;
- Отображение истории обработки.

### **2.3.16. Описание функций модуля контроля и учета вывоза**

Модуль контроля и учета вывоза включает в себя следующие функциональные блоки:

- Функциональный блок взаимодействия с мусоровывозящими организациями;
- Функциональный блок интеграции с внешними системами;
- Функциональный блок «Телематическая платформа».

### **2.3.17. ФБ Взаимодействия с мусоровывозящими организациями**

Функциональный блок взаимодействия с мусоровывозящими организациями включает в себя следующие функции:

- ведение договоров управления вывозом.
- работа с механизмами распределения зон деления возчиков.
- передача площадок и заявок возчикам, согласно зонам ответственности.
- сбор и визуализация информации по фактам вывоза возчиков.
- получение информации о ТС возчика.
- тестирование БНСО ТС возчика.
- согласование ТС возчика для последующего использования в вывозе заданий РО.
- контроль выполнения заданий возчиком.
- обеспечение дополнительных режимов контроля.

### **2.3.18. Функции ведения договоров управления вывозом**

Функции обеспечивают ведение договорных отношений между региональным оператором и организацией, осуществляющей сбор и транспортировку отходов.

Ведение договоров управления вывозом позволяет обеспечить:

- создание договора управления вывозом с формированием набора атрибутов договора, таких как:
  - идентификационные данные договора:
    - номер;
    - дата подписания;
    - срок действия.
  - данные о контрагентах договора:
    - наименование;
    - вид услуг/работ;
    - тип отходов;
    - вид договора;
    - прототипы;
    - транспортные средства;
    - объем вывоза;
    - перечень площадок.
- сопутствующие документы (сканы оригиналов сопутствующих документов);
- дополнительные соглашения.
- Редактирование договора управление вывозом;
- Удаление договора управления вывозом.

### **2.3.19. Функция работы с механизмами распределения зон деления возчиков**

Функция работы с механизмами распределения зон деления обеспечивают ведение реестра зон деления возчиков и включают в себя:

- Создание зоны деления;
- Редактирование зоны деления;
- Создание правила автоматической привязки новых площадок на основе зон деления;
- Перераспределение площадок на основе зон деления.

#### **2.3.20. Функция передачи площадок и заявок возчикам, согласно зонам ответственности**

Функция предназначен для автоматической передачи заявок и площадок организации, осуществляющей сбор и транспортировку отходов, в соответствии с ее зоной ответственности.

Функция передачи площадок и заявок возчикам обеспечивает:

- Автоматическую передачу площадок в систему организации, осуществляющей сбор и транспортировку отходов;
- Автоматическую передачу заявок по переданным площадкам в систему организации, осуществляющей сбор и транспортировку отходов.

#### **2.3.21. Функция сбора и визуализации информации по фактам вывоза возчиками**

Функция предназначена для сбора и визуального отображения результатов фактического выполнения работ по вывозу отходов возчиками.

Функция сбора и визуализации информации по фактам вывоза возчиками обеспечивает:

- Отображение результата выполнения работ по выполнению заявки;
- Отображение результата посещения площадки;
- Отображение фотографий, прикрепленных к результату выполнения заявки.

#### **2.3.22. Функция получения информации о ТС возчика**

Функция предназначена для получения достоверной информации о ТС, осуществляющем транспортировку отходов со стороны возчика.

Функция получения информации о ТС возчика обеспечивает:

- Получение информации о государственном номере ТС;
- Получение информации о модели ТС;
- Получение информации о БНСО;
- Получение информации о политике учета и посещений.

#### **2.3.23. Функция тестирования навесного оборудования ТС возчика**

Функция тестирования навесного оборудования предназначен для получения информации о наличии телематических данных ТС при получении заявки на согласование ТС.

Функция тестирования навесного оборудования ТС обеспечивает:

- Автоматическое тестирование наличия телематических данных ТС;

- Ручное тестирование наличия телематических данных ТС.

#### **2.3.24. Функция согласования ТС водителя для последующего использования в вывозе заданий РО**

Функция предназначена для согласования ТС на стороне регионального оператора.

Функцией согласования ТС водителя обеспечивается:

- Согласование ТС;
- Рассогласование ТС;
- Отображение краткой информации по ТС.

#### **2.3.25. Функция контроля выполнения заданий водителем**

Функция предназначена для обеспечения непрерывного контроля выполнения заданий по вывозу ТКО.

Функцией контроля выполнения заданий водителем обеспечивается:

- Контроль местоположения ТС;
- Контроль стоянок и остановок;
- Контроль срабатывания БНСО (навигационного терминала GPS/Глонасс);
- Отображение оперативной информации по данным контроля вывоза.

#### **2.3.26. Функция дополнительных режимов контроля**

Функция предназначена для осуществления дополнительного контроля посещения площадок организациями, осуществляющими сбор и транспортировку ТКО.

Функцией дополнительных режимов контроля (АСУ ОДС) обеспечивается:

- Отображение на карте посещений сведений о местоположении площадок.
- Отображение на карте посещений сведений о посещениях площадок транспортными средствами.

#### **2.3.27. Описание функций модуля аналитики и отчетности**

Модуль аналитики и отчетности включает в себя следующие функциональные блоки:

- ФБ формирования отчетов.
- ФБ аналитической отчетности.

#### **2.3.28. Функциональный блок формирования отчетов**

ФБ формирования отчетов предназначен для сбора, обработки и представления информации по функционированию отрасли обращения с отходами в различной форме и в различных разрезах.

ФБ формирования отчетов обеспечивает выполнение следующих функций:

- Формирование отчета;
- Экспорт отчета;

- Печать отчета.

Перечень отчетов представлен в Приложении 2.

### **2.3.29. Функциональный блок аналитической отчетности**

ФБ предназначен для расчета и отображения оперативных, аналитических и статистических данных в виде графиков, таблиц и диаграмм.

ФБ аналитической отчетности обеспечивает выполнение следующих функций:

- Формирование и графическое отображение тепловой карты заключения договоров на базе целевых показателей регионов.

### **2.3.30. Описание функций модуля пользовательского интерфейса**

Модуль пользовательского интерфейса обеспечивает непосредственный доступ пользователей к функциям АИС «Отходы». Доступ реализован по технологии «тонкого клиента» (через веб браузер) и предоставляет графический интерфейс автоматизированных рабочих мест, реализующий определенные наборы функций для следующих типов пользователей:

- сотрудник коммерческого отдела;
- заказчик вывоза ТКО;
- специалист;
- руководитель.

Для указанных типов пользователей модуль предоставляет следующие наборы функций:

#### **1) Функции АРМ Сотрудника коммерческого отдела:**

- Создание, редактирование и удаление заявок на вывоз ТКО;
- Создание и редактирование договоров с отходообразователями;
- Создание и редактирование договоров с организациями, осуществляющими вывоз ТКО;
- Создание и редактирование площадок;
- Создание и редактирование групп площадок;
- Создание и редактирование графиков вывоза отходов;
- Создание и редактирование информации о заказчиках вывоза ТКО;
- Согласование ТС, обеспечивающих вывоз;
- Контроль вывоза.

#### **2) Функции АРМ Заказчика вывоза ТКО:**

- Создание, редактирование и удаление заявок на вывоз;
- Просмотр статуса заявки;
- Подача корректировок по вывозу;

#### **3) Функции АРМ Специалиста:**

- Создание и редактирование площадок;

- Создание и редактирование групп площадок;
  - Создание и редактирование графиков вывоза отходов.
- 4) Функции АРМ Руководителя:
- Просмотр информации об исполнении договоров на вывоз ТКО;
  - Формирование отчетной и статистической информации.

### **2.3.31. Описание функций модуля администрирования**

Модуль администрирования включает в себя следующие функции:

- Управление системными настройками;
- Управление настройками телематической платформы;
- Управление настройками прав доступа.

При управлении настройками прав доступа хранение и обработка персональных данных в Системе не осуществляется.

### **2.3.32. Описание функций модуля аудита**

Основным назначением модуля аудита является фиксация основных событий в Системе для дальнейшего использования в разборе возможных инцидентов. Модуль не имеет интерфейса – доступ к информации о событиях осуществляется средствами системы управления базами данных.

Модуль аудита включает в себя следующие функции:

- Фиксация событий входа/выхода в систему.

### **3. ВЗАИМОСВЯЗЬ СО СМЕЖНЫМИ СИСТЕМАМИ**

#### **3.1. Общие сведения**

Функциональный блок интеграции данных включает в себя следующие функции:

- взаимодействие с системами телеметрии
- взаимодействие с внешними информационными системами ведения контрагентов и договоров.
- взаимодействие с внешней информационной системой оператора датчиков

#### **3.2. Функция телеметрии**

Функция телеметрии предназначена для сбора и хранения данных, поступающих с абонентских телематических терминалов, а также устройств и датчиков, подключенных к БНСО, в том числе:

- идентификационный номер абонентского терминала;
- географическая широта местоположения транспортного средства;
- географическая долгота местоположения транспортного средства;
- скорость движения транспортного средства;
- путевой угол транспортного средства;
- время и дата фиксации местоположения транспортного средства;

Телеметрия обеспечивает:

- Прием мониторинговой информации с БНСО.

Телеметрия поддерживает следующие протоколы:

- EGTS (Приказ №285 Министерства транспорта РФ «Об утверждении требований к средствам навигации, функционирующим с использованием навигационных сигналов системы ГЛОНАСС или ГЛОНАСС / GPS и предназначенным для обязательного оснащения транспортных средств категории М, используемых для коммерческих перевозок пассажиров, и категории N, используемых для перевозки опасных грузов»).

#### **3.3. Функция взаимодействия с внешними информационными системами ведения контрагентов, договоров и расчета заработной платы**

Функция обеспечивает интеграцию АИС «Отходы» с внешней информационной системой ведения контрагентов и договоров и включает в себя:

- Взаимодействие с информационной системой ведения контрагентов и договоров.

Реализуемые функции обеспечивают обмен следующими данными (в том числе, но не ограничиваясь):

- Договор;

- Вид контрагента;
- Контрагент;
- Площадка;
- Район;
- Вид отходов;
- Тип емкости;
- Показатели.

Информационный обмен с внешними системами осуществляется посредством специализированных веб сервисов с выполнением требований законодательства РФ в области защиты информации при использовании средств защиты информации, предоставляемых Заказчиком.

Детализированное описание сервисов приведено в Приложении № 1.

### **3.4. Функция взаимодействия с внешними информационными системами оператора датчиков**

Функция обеспечивает интеграцию АИС «Отходы» с внешней информационной системой контроля УЗ датчиков контейнеров и включает в себя:

- Взаимодействие с информационной системой контроля датчиков.

Реализуемые функции обеспечивают обмен следующими данными (в том числе, но не ограничиваясь):

- Контейнерная группа;
- Площадка;
- Процент заполнения.

Информационный обмен с внешними системами осуществляется посредством специализированных веб сервисов с выполнением требований законодательства РФ в области защиты информации при использовании средств защиты информации, предоставляемых Заказчиком.

Детализированное описание сервисов приведено в Приложении № 2.

### Описание информационного взаимодействия с внешней системой в части передачи информации по договорам и площадкам

В настоящем приложении описаны процессы обмена информацией между АИС «Отходы» и внешней системой в части передачи информации по договорам и площадкам, регистрируемым в учетной системе предприятия.

Информационный обмен осуществляется посредством специализированного Web-сервиса. Описание модели данных, используемой сервисом, а также доступных методов представлено в WSDL-описании, доступным по ссылкам:

<https://kro.tko-inform.ru/ro/services/customers?wsdl>

<https://kro.tko-inform.ru/ro/services/staffs?wsdl>

В рамках интеграции реализуется следующая модель взаимодействия и структуры обмена данными:



Порядок обмена информацией, на примере нового контрагента:

1. Коммерческая служба добавляет нового контрагента во внешней системе.  
→ Данные по новому контрагенту выгружаются в АИС Отходы
2. Коммерческая служба добавляет договор, указывая его параметры, а также указывая на какие площадки распространяется договор.

Если договор «по норме», то привязка выполняется к уже существующей площадке. Если договор «по факту», то привязка осуществляется или к существующей площадке или коммерческий отдел создает прототип площадки, указывая исходя из данных договора наименование площадки, виды отходов, типы емкостей и их количество.

→ Данные по новому договору выгружаются в АИС Отходы. Для договоров «по норме» на соответствующей площадке добавляется соответствующий отходообразователь.

3. По факту получения данных по договорам, на основании данных прототипов площадок, служба эксплуатации Предприятия заводит фактические стационарные площадки в АИС Отходы или обновляет данные на существующей (добавляет новую контейнерную группу)

→ Данные по площадке выгружаются во внешнюю систему.

### **Передача информации о продлении договора**

АИС «Отходы» позволяет продлить дату завершения существующего в системе договора. В данном случае при получении обновления из внешней системы дата завершения действующего договора увеличивается. Дата завершения договора обновляется в комплексном типе `<xs:complexType name="Contract">` элемент `<xs:element name="ExpirationDate" type="xs:date"/>`.

### **Передача информации о расторжении договора**

Информация о расторжении договора между Customer и Supplier приходит в АИС «Отходы» как обновление существующего в системе договора. Дата завершения договора обновляется в комплексном типе `<xs:complexType name="Contract">` элемент `<xs:element name="ExpirationDate" type="xs:date"/>`.

### **Передача дополнительного соглашения по договору**

Дополнительное соглашение приходит в АИС «Отходы-Региональный оператор» в виде нового договора. Для идентификации того, что данный договор является продлением существующего в системе договора, должен быть заполнен комплексный тип `<xs:complexType name="Contract">` элемент `<xs:element name="ContractRef" type="tns:ContractRef"/>`.

### **Обновление информации по площадке в рамках договора**

При регистрации изменений информации по площадке во внешней системе в рамках договора необходимо пересылать обновленный договор в АИС «Отходы-Региональный оператор» . Обновление происходит по идентификатору договора `<xs:complexType name="Contract">` сначала по полю `<xs:element name="Id" type="xs:string"/>`, если оно не заполнено, то по полю `<xs:element name="Reference" type="xs:string"/>`, где:

- name="Id" – идентификатор договора в АИС «Отходы-Региональный оператор» ;

- name="Reference" – идентификатор договора в учетную систему предприятия.

### **Добавление новой площадки к существующему договору**

При добавлении новой площадки к существующему договору из внешней системы необходимо пересылать в АИС «Отходы-Региональный оператор» обновленный договор, содержащий весь набор площадок по данному договору.

### **Удаление площадки в договоре**

При удалении площадки в договоре из внешней системы необходимо пересылать в АИС «Отходы-Региональный оператор» обновленный договор, содержащий обновленный набор площадок, в котором отсутствует удаленная площадка.

При регистрации изменений информации по площадке во внешней системе в рамках договора необходимо пересылать обновленный договор в АИС «Отходы-Региональный оператор» . Обновление происходит по идентификатору договора `<xs:complexType name="Contract">` сначала по полю `<xs:element name="Id" type="xs:string"/>`, если оно не заполнено, то по полю `<xs:element name="Reference" type="xs:string"/>`, где:

- name="Id" – идентификатор договора в АИС «Отходы-Региональный оператор» ;
- name="Reference" – идентификатор договора в учетной системе предприятия.

### **Добавление новой площадки к существующему договору**

При добавлении новой площадки к существующему договору из внешней системы необходимо пересылать в АИС «Отходы-Региональный оператор» обновленный договор, содержащий весь набор площадок по данному договору.

### **Удаление площадки в договоре**

При удалении площадки в договоре из внешней системы необходимо пересылать в АИС «Отходы-Региональный оператор» обновленный договор, содержащий обновленный набор площадок, в котором отсутствует удаленная площадка.

## Описание Public API системы контроля датчиков

Функции Public API можно вызвать, используя следующую спецификацию.

```
<protocol>:<host>/web/public/api/<имя функции>?<параметры запроса>
```

Функции возвращают данные в виде JSON. Обязательным полем объекта является поле «ver». Его значение равно номеру текущей версии WPAPI.

### Функция обеспечивающая доступ ко всей функциональности WPAPI

```
gettoken?login=<знач>&pswd=<знач>&sysid=<знач> [&renew=1]  
Метод HTTP-запроса: GET
```

Параметры:

login- логин пользователя;

pswd- пароль;

sysid- уникальный идентификатор внешней системы.

renew- [опционно] принудительный выпуск нового токена.

Результат:

Ответ в виде объекта {"token": <знач>}, или если ошибка {"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - Пользователь или пароль неверен.

2 - Проблемы с алгоритмом хеширования.

3 - Неверный идентификатор внешней информационной системы,

500 – внутренняя ошибка на сервере

### Функция, возвращающая информацию об операторе и доступных пользователю областях обслуживания

Если пользователем, которому был выдан токен, является администратор оператора, то доступны все области обслуживания данного оператора.

```
getinfo?token=<знач>  
Метод HTTP-запроса: GET
```

Параметры:

token- токен выданный методом gettoken

Результат:

Ответ в виде объекта {"holder":{..},"availableareas":[{..},..]}.

Структура поля holder:

Поле	Тип	Описание
id	int	Идентификатор оператора
active	int	Флаг активности
address	string	Адрес
code	string	Код оператора
dbaFlag	int	Зарезервировано
shortName	string	Короткое имя оператора
fullName	string	Полное имя оператора
host	string	IP адрес обрабатывающего хоста
okpo	string	Код ОКПО оператора
si	string	Идентификатор внешней системы
tel	string	Номер телефона оператора

Поле availableareas является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор области обслуживания
code	string	Код области обслуживания
name	string	Имя области обслуживания
fullName	string	Полное имя области обслуживания
latitude	float	Широта центра области обслуживания
longitude	float	Долгота центра области обслуживания
timeZone	string	Часовой пояс области обслуживания
vendorMap	string	Название картографического сервиса.
holder	object	Оператор

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида  
{"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### **Функции управления нормативно-справочной информацией**

**enumdrivers?token=<знач>**

**Метод HTTP-запроса: GET**

Возвращает список зарегистрированных водителей.

Параметры:

token- токен выданный функцией gettoken

Результат:

Ответ в виде объекта {"drivers":[{"..},...]}. Поле drivers является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор водителя
name	string	Имя водителя
surname	string	Фамилия водителя
mname	string	Отчество водителя
fullName	string	Полное ФИО водителя
licenseNumber	string	Номер водительского удостоверения
licenseCategory	string	Категория прав
tabelNumber	string	Табельный номер

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

**Функция, выполняющая модификацию справочника водителей по переданному в теле запроса массиву**

```
putdriver?token=<знач>& areaId=<знач> [&mergeto=<знач>]
```

Метод HTTP-запроса: POST

Если у переданном водителе существует непустое поле id, то происходит поиск и обновление данных (если единица найдена), иначе вставка новой записи. При

использовании параметра mergeto тоже самое будет выполняться со списком полей, указанных в mergeto. Если по результат поиска с полями mergeto найдено более одной записи, обновление данных произойдет для всех найденных водителей.

Параметры:

token - токен выданный методом gettoken areaId– идентификатор области обслуживания

mergeto– [опционально] список имен полей, через запятую

Тело запроса

Содержит JSON. Он содержит массив объектов со следующими полями:

Поле	Тип	Описание
------	-----	----------

id	int	Идентификатор водителя
name	string	Имя водителя
surname	string	Фамилия водителя
mname	string	Отчество водителя
licenseNumber	string	Номер водительского удостоверения
licenseCategory	string	Категория прав
tabelNumber	string	Табельный номер

Результат:

Ответ в виде объекта вида {"inserted":[21,5,..], "updated":[17,..], "ignored":[1,..]}. В массивах возвращаются идентификаторы записей, которые, соответственно, были вставлены, обновлены или проигнорированы. Игнорируются записи в связи с отсутствием в базе данных. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}. Возможные коды ошибок:

- 1 - неверный или устаревший токен,
- 500 – внутренняя ошибка на сервере

#### Функция, возвращающая список контрагентов

```
enumcompanys?token=<знач> [&active=<знач>]
```

Метод HTTP-запроса: GET

Параметры:

token - токен выданный методом gettoken active - [опционно] Фильтр. Возможные значения:

- 0 - Только не активных 1 - Только активных
- 2 - Выбрать всех (по умолчанию)

Результат:

Ответ в виде объекта {"companys":[{"..},..]}. Поле companys является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор контрагента
name	string	Короткое имя
fullName	string	Полное имя
inn	string	ИНН
kpp	string	КПП
ogrn	string	ОГРН
addressLegal	string	Адрес юридический
addressFact	string	Адрес фактический

contactName	string	Имя контактного лица
contactTel	string	Номер телефона контактного лица
contactEmail	string	Почта контактного лица
contractNumber	string	Номера действующих договоров
contractDate	long	Дата заключения первого договора
active	int	Флаг активности. 1 – активный контрагент, 0 – контрагент имеет архивный статус

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### Функция, возвращающая контрагентов по их идентификаторам

```
getcompany?token=<знач>&ids=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken ids- список идентификаторов, через запятую

Результат:

Ответ в виде объекта {"companys":[{"..},,..]}. Структура companys указана в описании **enumcompanys**. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### Функция, выполняющая модификацию справочника контрагентов по переданному в теле запроса массиву контрагентов

```
putcompany?token=<знач> [&mergeto=<знач>]
```

Метод HTTP-запроса: POST

Если у контрагента существует непустое поле id, то происходит поиск и обновление данных контрагента (если контрагент найден), иначе вставка новой записи. При использовании параметра mergeto тоже самое будет выполняться со списком полей, указанных в mergeto. Если по результат поиска с полями mergeto найдено более одной записи, обновление данных произойдет для всех найденных контрагентов.

Параметры:

token- токен выданный методом gettoken

mergeto– [опционно] список имен полей, через запятую

Тело запроса

Содержит JSON. Он содержит массив объектов со следующими полями\*:

Поле	Тип	Описание
id	int	Идентификатор контрагента
name	string	Короткое имя
fullName	string	Полное имя
inn	string	ИНН
kpp	string	КПП
buc	string	Код бизнес единицы
ogrn	string	ОГРН
addressLegal	string	Адрес юридический
addressFact	string	Адрес фактический
contactName	string	Имя контактного лица
contactTel	string	Номер телефона контактного лица
contactEmail	string	Почта контактного лица
contractNumber	string	Номера действующих договоров
contractDate	long	Дата заключения первого договора
active	int	Флаг активности. 1 – активный контрагент, 0 – контрагент имеет архивный статус

Результат:

Ответ в виде объекта вида {"inserted":[21,5,..], "updated":[17,..], "ignored":[1,..]}. В массивах возвращаются идентификаторы записей, которые, соответственно, были вставлены, обновлены или проигнорированы. Игнорируются записи в связи с отсутствием в базе данных. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}. Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

**Функция, возвращающая номенклатурный список**

```
enumcargoes?token=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken

Результат:

Ответ в виде объекта {"cargoes":[{},..]}. Поле cargoes является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор номенклатурной единицы
name	string	Имя номенклатурной единицы
fkko	string	Код по ФККО
dangerclass	int	Класс опасности
smwFlag	int	Флаг ТКО

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида  
{"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### **Функция, возвращающая список номенклатурных единиц по их идентификаторам**

```
getcargo?token=<знач>&ids=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken ids- список идентификаторов, через запятую

Результат:

Ответ в виде объекта {" cargoes":[ {...},...]}]. Структура cargoes указана в описании **enumcargoes**. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### **Функция, выполняющая модификацию справочника номенклатурных единиц по переданному в теле запроса массиву**

```
putcargo?token=<знач> [&mergeto=<знач>]
```

Метод HTTP-запроса: POST

Если у переданной номенклатурной единицы существует непустое поле id, то происходит поиск и обновление данных (если единица найдена), иначе вставка новой записи. При использовании параметра mergeto тоже самое будет выполняться со списком полей, указанных в mergeto. Если по результат поиска с полями mergeto найдено более одной записи, обновление данных произойдет для всех найденных номенклатурных единиц.

Параметры:

token- токен выданный методом gettoken

mergeto– [опционно] список имен полей, через запятую

Тело запроса

Содержит JSON. Он содержит массив объектов со следующими полями:

Поле	Тип	Описание
id	int	Идентификатор номенклатурной единицы
name	string	Имя номенклатурной единицы
fkko	string	Код по ФККО
dangerclass	int	Класс опасности

Результат:

Ответ в виде объекта вида {"inserted":[21,5,...], "updated":[17,...], "ignored":[1,..]}. В массивах возвращаются идентификаторы записей, которые, соответственно, были вставлены, обновлены или проигнорированы. Игнорируются записи в связи с отсутствием в базе данных. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}. Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

#### Функция, возвращающая список транспортных средств

```
enumtrucks?token=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken

Результат:

Ответ в виде объекта {"trucks":[{"..},...]}. Поле companys является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор транспортного средства (ТС)
kind	object	Модель
id	int	Идентификатор модели ТС
capacity	int	Вместимость кузова, в куб.см.
compactionRatio	float	Степень уплотнения (для спецтехники)
model	string	Название модели
vehicleType	object	Тип ТС
loadingMethod	object	Метод загрузки
fuelConsumption	int	Расход топлива в л/100 км.
weight	int	Грузоподъемность (в кг.)
fullName	string	Полное имя ТС

Поле	Тип	Описание
parking	object	Парковка к которой ТС приписано
plate	string	Гос.номер
driver	object	Водитель основной
drivers	Array of object	Водители сменные
specConditions	int	Число, содержащее битовые флаги: 0 бит – Арендованное ТС
fuelConsumption		Расход топлива ТС в л/100 км.
compactionRatio		Степень уплотнения ТС (для спецтехники)
active	int	Флаг активности. 1 – активное ТС, 0 –ТС имеет архивный статус

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

**Функция, возвращающая список транспортных средств по их идентификаторам**

```
gettruck?token=<знач>&ids=<знач>
```

Метод HTTP-запроса: GET

**Параметры:**

token- токен выданный методом gettoken ids- список идентификаторов, через запятую

Результат:

Ответ в виде объекта {"trucks":[{"..},..]}. Структура trucks указана в описании **enumtrucks**. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

**Функция, выполняющая модификацию справочника транспортных средств (ТС) по переданному в теле запроса массиву**

```
puttruck?token=<знач>&parkingId=<знач> [&mergeto=<знач>]
```

Метод HTTP-запроса: POST

Если у переданного ТС существует непустое поле id, то происходит поиск и обновление данных (если единица найдена), иначе вставка новой записи. При использовании параметра

mergeto тоже самое будет выполняться со списком полей, указанных в mergeto. Если по результат поиска с полями mergeto найдено более одной записи, обновление данных произойдет для всех найденных ТС.

Параметры:

token- токен выданный методом gettoken parkingId– идентификатор парковки

mergeto– [опционно] список имен полей, через запятую

Тело запроса

Содержит JSON. Он содержит массив объектов со следующими полями:

Поле	Тип	Описание
id	int	Идентификатор номенклатурной единицы
kind	int	Идентификатор модели
parking	object	Парковка к которой ТС приписано
plate	string	Гос. номер
driver	int	Идентификатор основного водителя
drivers	array of int	Идентификатор сменных водителей
specConditions	int	Число, содержащее битовые флаги: 0 бит – Арендованное ТС
fuelConsumption	int	Расход топлива ТС в л/100 км.
compactionRatio	float	Степень уплотнения ТС (для спецтехники)
active	int	Флаг активности. 1 – активное ТС, 0 –ТС имеет архивный статус

Результат:

Ответ в виде объекта вида {"inserted":[21,5,..], "updated":[17,..], "ignored":[1,..]}. В массивах возвращаются идентификаторы записей, которые, соответственно, были вставлены, обновлены или проигнорированы. Игнорируются записи в связи с отсутствием в базе данных. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### **Функция, возвращающая список групп планирования(участков)**

```
enumgroups?token=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный функцией gettoken.

Результат:

Ответ в виде объекта {"groups":[{"..},..]}: [{"..},..]}. Поле groups

является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор группы планирования
name	string	Название группы планирования. Например, «Центральный район»
fullName	string	Полное название группы планирования. Например, «Центральный район, Пермь, Экогород»
usingForFinalPlanning	int	Флаг использование группы для завершающего планирования.

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### Функция, возвращающая список нештатных ситуаций

`enumunexproute?token=<знач>`

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken

Результат:

Ответ в виде объекта {"unexproute":[{"..},..]}. Поле unexproute является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор
reason	string	Причина
comment	string	О причине подробно

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

**Функция, возвращающая список мест накопления (площадок) в группе планирования**

```
enumpoints?token=<знач>&groupId=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token - токен выданный методом gettoken, groupId - идентификатор группы планирования.

Результат:

Ответ в виде объекта {"points":[{"..},..]}. . Поле points является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор места накопления
address	string	Адрес площадки
areaCode	string	Код области обслуживания
areaName	string	Имя области обслуживания
groupId	int	Идентификатор группы планирования или null
groupName	string	Имя группы планирования или null
kind	string	Тип площадки. Одно из: 'SENSOR', 'MANUAL' или 'CONSTANT'.
latitude	float	Широта
longitude	float	Долгота
proposalId	int	Идентификатор заявки или null
calc	object	
active	bool	Активность площадки
alarmed	bool	Один из контейнеров на площадке передал сигнал о тревоге «Пожар»
broken	bool	Один из контейнеров на площадке не выходит на связь более суток (+2 часа)

Поле	Тип	Описание
avgFillingRate	float	Средняя скорость накопления отходов на площадке (куб.см./сут.)
containerCount	int	Кол-во активных контейнеров на площадке (тех, которые находятся во включенном режиме)
color	string	Цвет индикатора в hex
fullnessPercent	int	Фактическая наполненность площадки в процентах на момент запроса
indexQuality	float	Критерий качества уборки. Представляет собой среднее кол-во суток, когда площадка находилась в полностью заполненном состоянии. Чем выше данная величина, тем хуже качество уборки
lastFilled	long	Дата/время последнего перехода в наполненное состояние контейнера площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
lastMessage	long	Дата/время последнего сообщения от прибора площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
lastVisit	long	Дата/время последнего опорожнения контейнеров площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
scheduledTimeFilling	long	Дата/время планируемого заполнения площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
scheduledVolume	int	Расчетная наполненность площадки на момент запроса (в куб.см.)
volume	int	Суммарный объем контейнеров на площадке (в куб.см.)
selectiveContainerCount	int	Количество контейнеров на площадке для отдельного сбора

Поле	Тип	Описание
selectiveCalculatePoints	object	Мар объект, ключом которого является идентификатор вида фракции, а значением – объект типа calc

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

#### **Функция, возвращающая информацию о месте накопления (площадке)**

```
getpoint?token=<знач>&pointId=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken, pointId- идентификатор места накопления.

Результат:

Ответ в виде объекта {"point":{..}}. Структура объекта point аналогична структуре, описанной в функции enumpoints.

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида {"errcode":<код ошибки>,"message":<сообщение об ошибке>}.

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

#### **Функция, возвращающая информацию о местах накопления (площадке), расположенных поблизости от заданных координат**

```
getpointsbycoord?token=<знач>&coord=<знач>[&radius=<знач>]
```

Метод HTTP-запроса: GET

Параметры:

token - токен выданный методом gettoken; coord- координаты, заданные в формате, lat,lng;

radius - [опционально] радиус окрестности в метрах, в которой нужно искать площадки.

Если параметр пропущен, площадки ищутся в окружности радиусом 260 метров

Результат:

Ответ в виде объекта {"points":[{"..},...]}. . Поле points является массивом. Структура объекта points аналогична структуре, описанной в функции enumpoints. Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида

```
{"errcode":<код ошибки>,"message":<сообщение об ошибке>}
```

Возможные коды ошибок:

1 - неверный или устаревший токен,

500 – внутренняя ошибка на сервере

### Функция, возвращающая контейнеры, установленные на площадке

```
enumcontainers?token=<знач>&pointId=<знач>
```

Метод HTTP-запроса: GET

Параметры:

token - токен выданный методом gettoken, pointId- идентификатор места накопления (площадки).

Результат:

Ответ в виде объекта {"containers":[{"..},...]}. . Поле containers является массивом, состоящим из элементов, имеющих следующую структуру:

Поле	Тип	Описание
id	int	Идентификатор контейнера
code	string	Код контейнера
lastMessage	long	Дата/время последнего сообщения от прибора площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
lastVisit	long	Дата/время последнего опорожнения контейнеров площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
lastFilled	long	Дата/время последнего перехода в наполненное состояние контейнера площадки (количество миллисекунд, начиная с 1 января 1970 года, UTC)
numberOfMeasure	int	Кол-во циклов измерений
fillingRate	float	Скорость накопления (куб.см./сут.)
pointId	int	Идентификатор площадки
state	string	Состояние контейнера. Одно из: 'READY', 'FILLED', 'ALARM', 'ERROR', 'INACTIVE'
type	object	Тип контейнера
id	int	Идентификатор типа

Поле	Тип	Описание
code	string	Код типа
volume	int	Вместимость (куб.см.)
bunkerFlag	int	Флаг перевозимого бункера
extraTime	int	Среднее время (в миллисекундах), которое контейнер находился в заполненном состоянии
subscriberNumber	string	Номер абонента
kindSMW	object	Фракция
id	int	Идентификатор фракции
name	string	Наименование фракции
endOfServicePeriod	long	Дата/время окончания обслуживания (количество миллисекунд, начиная с 1 января 1970 года, UTC)
typeDev	string	Тип коммуникаций. Одно из: 'GSM', 'LORAWAN'
ttc_empty	int	Кол-во интервалов для подтверждения опорожнения. Размер интервала – 5 мин. Если null действует аппаратное умолчание(зависит от версии прошивки).
ttc_filled	int	Кол-во интервалов для подтверждения заполнения. Размер интервала – 5 мин. Если null действует аппаратное умолчание (зависит от версии прошивки).
firmwareVersion	int	Версия прошивки
measuringMethod	string	Тип измерения. Одно из: 'WASTEOUT_METHOD', 'LEVEL_METHOD'
suspensionDistance	int	Расстояние подвеса (в мм). Имеет смысл для measuringMethod = MeasuringMethod.LEVEL_METHOD. Для измерения уровня прибор может подвешиваться над контейнером, тогда это значение должно вычитаться из эталонного расстояния
level	string	Уровень в процентах. Имеет смысл для measuringMethod = MeasuringMethod.LEVEL_METHOD

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида  
{"errcode":<код ошибки>,"message":<сообщение об ошибке>}

Возможные коды ошибок:

- 1 - неверный или устаревший токен,
- 2 - площадка не найдена,
- 500 – внутренняя ошибка на сервере

**Функция, возвращающая фактическое местонахождение контейнера по информации о базовых станциях сотовой сети (только для GSM и NBioT приборов), которые были известны прибору при выходе на связь в момент времени date.**

```
getlocation?token=<знач>&containerId=<знач> [&date=<знач>]
```

Метод HTTP-запроса: GET

Параметры:

token- токен выданный методом gettoken, containerId- идентификатор контейнера,

date – [опционно] дата выхода прибора на связь в виде количества

миллисекунд, начиная с 1 января 1970 года, UTC. Если дата выхода на связь не известна, она может быть не указана. В этом случае будет выбрана последняя дата/время, когда прибор выходил на связь.

Результат:

Ответ в виде объекта с полями

Поле	Тип	Описание
bsCount	int	Кол-во базовых станций
date	long	Дата/время выхода на связь в виде количество миллисекунд, начиная с 1 января 1970 года, UTC
lbs	object	Объект с вариантами местонахождения
yandex		По данным LBS «Яндекс.Локатор». Массив содержат три вещественных числа, определяющих широту, долготу и точность в метрах соответственно.
ocid		По данным LBS «OpenCellId». Массив содержат три вещественных числа, определяющих широту, долготу и точность в метрах соответственно.
google		По данным LBS «Google GeoLocation». Массив содержат три вещественных числа, определяющих широту, долготу и точность в метрах соответственно.

В дальнейшем, по мере подключения других LBS, в поле “lbs” могут появляться другие варианты. При выборе подходящего варианта рекомендуется опираться на точность измерения.

Если возникла ошибка обработки запроса, то будет возвращен GSON-ответ вида

```
{"errcode":<код ошибки>,"message":<сообщение об ошибке>}
```

Возможные коды ошибок:

1 - неверный или устаревший токен,

5 - контейнер не найден

6 - информация о базовых станциях не найдена

7 - ошибка LBS

500 – внутренняя ошибка на сервере